Using Network Printers

This help document describes how to discover, configure, and use TCP/IP network printers with CUPS.

Getting the IP Address

Every network printer or print server has a unique Internet Protocol (IP) address associated with it. This address is either configured manually or set using an automatic network protocol such as the Boot Protocol (BOOTP), Dynamic Host Control Protocol (DHCP), Reverse Address Resolution Protocol (RARP), or ZeroConf.

You can normally find the IP address of a printer on the printer's control panel or by printing the configuration or status page. The Simple Network Management Protocol (SNMP) can also be used to get the IP address remotely, which happens automatically when you visit the CUPS administration web page or choose an available CUPS device when adding a printer.

Configuring the IP Address

When you first install a network printer or print server on your LAN, you need to set the Internet Protocol ("IP") address. Most higher-end "workgroup" printers allow you to set the address through the printer control panel. However, if you have many printers you will want to assign the addresses remotely - this makes administration a bit easier and avoids assigning duplicate addresses accidentally.

To setup your printer or print server for remote address assignment, you'll need the Ethernet Media Access Control ("MAC") address, also sometimes called a node address, and the IP address you want to use for the device. The Ethernet MAC address can often be found on the printer test page or bottom of the print server.

Configuring the IP Address Using DHCP

The DHCP protocol is the usual way of setting the IP address of a printer on a managed network. Using the standard `dhcpd(8)` program supplied with UNIX you simply need to add a line to the */etc/dhcpd.conf* file:

```
host hostname {
  hardware ethernet mac-address;
  fixed-address ip-address;
}
```

Make sure that the hostname you use is also listed in the */etc/hosts* file or is registered with your DNS server.

Configuring the IP Address Using BOOTP

The BOOTP protocol is used when you need to provide additional information such as the location of a configuration file to the network interface. Using the standard `bootpd(8)` program supplied with UNIX you simply need to add a line to the */etc/bootptab* file; for IRIX:

```
myprinter 08:00:69:00:12:34 192.0.2.2 myprinter.boot
```

Newer versions of `bootpd` use a different format:

```
myprinter:ha=080069001234:ip=192.0.2.2:t144=myprinter.boot
```

The *myprinter.boot* file resides in the */usr/local/boot* directory by default. If you do not need to provide a boot file you may leave the last part of the line blank.

Note: Some versions of UNIX do not enable the BOOTP service by default. The */etc/inetd.conf* or */etc/xinetd.d/bootp* file usually contains a line for the BOOTP service that can be uncommented if needed.

Verifying the Printer Connection

To test that the IP address has been successfully assigned and that the printer is properly connected to your LAN, type:

```
ping ip-address
```

If the connection is working properly you will see something like:

```
ping myprinter
PING myprinter (192.0.2.2): 56 data bytes
64 bytes from 192.0.2.2: icmp_seq=0 ttl=15 time=5 ms
64 bytes from 192.0.2.2: icmp_seq=1 ttl=15 time=3 ms
64 bytes from 192.0.2.2: icmp_seq=2 ttl=15 time=3 ms
64 bytes from 192.0.2.2: icmp_seq=3 ttl=15 time=3 ms
```

If not, verify that the printer or print server is connected to the LAN, it is powered on, the LAN cabling is good, and the IP address is set correctly. You can usually see the current IP address and network status by printing a configuration or test page on the device.

Network Protocols Supported by CUPS

CUPS supports most network printers using one of three TCP/IP-based protocols. Printer discovery is currently accomplished using the SNMP protocol, however future versions of CUPS will also include support for multicast DNS service discovery as well.

AppSocket Protocol

The AppSocket protocol (sometimes also called the JetDirect protocol, owing to its origins with the HP JetDirect network interfaces) is the simplest, fastest, and generally the most reliable network protocol used for printers. AppSocket printing normally happens over port 9100 and uses the `socket` URI scheme:

```
socket://ip-address-or-hostname
socket://ip-address-or-hostname?waiteof=false
socket://ip-address-or-hostname:port-number
socket://ip-address-or-hostname:port-number?waiteof=false
```

The "waiteof" option controls whether the `socket` backend waits for the printer to complete the printing of the job. The default is to wait.

Internet Printing Protocol (IPP)

IPP is the only protocol that CUPS supports natively and is supported by some network printers and print servers. However, since many printers do not implement IPP properly, only use IPP when the vendor actually documents official support for it. IPP printing normally happens over port 631 and uses the `http` and `ipp` URI schemes:

```
http://ip-address-or-hostname:port-number/resource
http://ip-address-or-hostname:port-number/resource?option=value
http://ip-address-or-hostname:port-number/resource?option=value&option=value
ipp://ip-address-or-hostname/resource
```

```
ipp://ip-address-or-hostname/resource?option=value
ipp://ip-address-or-hostname/resource?option=value&option=value
ipp://ip-address-or-hostname:port-number/resource
ipp://ip-address-or-hostname:port-number/resource?option=value
ipp://ip-address-or-hostname:port-number/resource?option=value&option=value
```

The `ipp` backend supports many options, which are summarized in Table 2.

Table 2: IPP URI Options

Option	Description
`compression=gzip`	Specifies that print data should be compressed before sending.
`encryption=always`	Specifies that the connection to the IPP server should be encrypted using SSL.
`encryption=ifrequested`	Specifies that the connection to the IPP server should only be encrypted if the server requests it.
`encryption=never`	Specifies that the connection to the IPP server should not be encrypted.
`encryption=required`	Specifies that the connection to the IPP server should be encrypted using TLS.
`version=1.0`	Specifies that version 1.0 of the IPP protocol should be used instead of the default version 1.1.
`waitjob=false`	Specifies that the IPP backend should not wait for the job to complete.
`waitprinter=false`	Specifies that the IPP backend should not wait for the printer to become idle before sending the print job.

Line Printer Daemon (LPD) Protocol

LPD is the original network printing protocol and is supported by many network printers. Due to limitations in the LPD protocol, we do not recommend using it if the printer or server supports one of the other protocols. LPD printing normally happens over port 515 and uses the `lpd` URI scheme:

```
lpd://ip-address-or-hostname/queue
lpd://ip-address-or-hostname/queue?option=value
lpd://ip-address-or-hostname/queue?option=value&option=value
```

Table 3 summarizes the options supported by the `lpd` backend.

Table 3: LPD URI Options

Option	Description
`banner=on`	Specifies that a banner page should be printed by the server.
`contimeout=`*`seconds`*	Specifies the number of seconds to wait for the connection to the server to complete.
`format=c`	Specifies that the print data is a CIF file.
`format=d`	Specifies that the print data is a DVI file.
`format=f`	Specifies that the print data is a plain text file.
`format=g`	Specifies that the print data is a Berkeley plot file.
`format=l`	Specifies that the print data is a raw (preformatted) print file.
`format=n`	Specifies that the print data is a ditroff file.
`format=o`	Specifies that the print data is a PostScript file.
`format=p`	Specifies that the print data is a plain text file that should be "pretty" printed with a header and footer.

`format=r`	Specifies that the print data is a FORTRAN carriage control file.
`format=t`	Specifies that the print data is a troff Graphic Systems C/A/T phototypesetter file.
`format=v`	Specifies that the print data is a Sun raster file.
`order=data,control`	Specifies that the print data files should be sent before the control file.
`reserve=none`	Specifies that the backend should not reserve a source port.
`reserve=rfc1179`	Specifies that the backend should reserve a source port from 721 to 731 as required by RFC 1179.
`sanitize_title=no`	Specifies that the job title string should not be restricted to ASCII characters.
`sanitize_title=yes`	Specifies that the job title string should be restricted to ASCII characters.
`timeout=`*`seconds`*	Specifies the number of seconds to wait for LPD commands to complete.

Common Network Printer URIs

Once you have set the IP address you can access the printer or print server using the `ipp`, `lpd`, or `socket` backends. Table 1 shows a list of common network interfaces and printer servers and the settings you should use with CUPS:

Table 1: Common Device URIs

Model/Manufacturer	Device URI(s)
Apple LaserWriter	lpd://*address*/PASSTHRU
Axis w/o IPP Axis OfficeBasic (see directions)	socket://*address*:9100 socket://*address*:9101 socket://*address*:9102
Axis w/IPP	ipp://*address*/LPT1 ipp://*address*/LPT2 ipp://*address*/COM1
Castelle LANpress™	lpd://*address*/pr1 lpd://*address*/pr2 lpd://*address*/pr3
DPI NETPrint	lpd://*address*/pr1 lpd://*address*/pr2 lpd://*address*/pr3
DLink DP-301P+	socket://*address*
EFI® Fiery® RIP	lpd://*address*/print
EPSON® Multiprotocol Ethernet Interface Board	socket://*address*
Extended System ExtendNET	lpd://*address*/pr1 lpd://*address*/pr2 lpd://*address*/pr3
Hewlett Packard JetDirect	socket://*address*:9100 socket://*address*:9101 socket://*address*:9102
Intel® NetportExpress XL, PRO/100	lpd://*address*/LPT1_PASSTHRU lpd://*address*/LPT2_PASSTHRU lpd://*address*/COM1_PASSTHRU
Lexmark™ MarkNet	lpd://*address*/ps
Linksys EtherFast® (see directions)	socket://*address*:4010 socket://*address*:4020 socket://*address*:4030
Linksys PSUS4	lpd://*address*/lp

Kodak®	lpd://*address*/ps
QMS® CrownNet™	lpd://*address*/ps
Tektronix® PhaserShare™	socket://*address*:9100
XEROX® 4512 NIC	lpd://*address*/PORT1
XEROX® XNIC	lpd://*address*/PASSTHRU
XEROX® (most others)	socket://*address*:5503

Troubleshooting SNMP Discovery Problems

Whenever you view the administration web page or a list of supported device URIs, the `snmp` backend will probe the local network(s) using Simple Network Management Protocol (SNMP) broadcasts. Printers that respond to these broadcasts are then interrogated for the make and model and supported protocols, yielding a device URI that can be used to add the printer.

That said, the SNMP requests sometimes expose problems in vendor SNMP or IPP implementations. If you are experiencing long delays in loading the CUPS web interface administration page, or if you don't see your printer listed, the following instructions will help you to diagnose those problems and/or provide important feedback to the CUPS developers so that we can correct problems and improve the SNMP backend in future releases.

Quick Fixes

If you don't use "public" as your community name, create a text file called */etc/cups/snmp.conf* and put the following line in it:

```
Community your community name
```

If you have more than one community name, list them all on separate lines.

If you don't support SNMP v1 on your network, you are currently "out of luck". That said, we will be adding v2, v2c, and v3 support in future CUPS releases once we have a handle on the actual requirements people have for such things. Please file or update an SNMP enhancement request with *specific* requirements you have - what you need supported, why you need it supported, and how you would like to see the functionality provided/exposed - so that we can do it "right" the first time.

Basic Debugging

The SNMP backend supports a debugging mode that is activated by running it from a shell prompt. If you are using Bash (/bin/bash), Bourne shell (/bin/sh), Korn shell (/bin/ksh), or Z shell (/bin/zsh), you can run the following command to get a verbose log of the SNMP backend:

```
CUPS_DEBUG_LEVEL=2 /usr/lib/cups/backend/snmp 2>&1 | tee snmp.log
```

For C shell (/bin/csh) and TCsh (/bin/tcsh), use the following command instead:

```
(setenv CUPS_DEBUG_LEVEL 2; /usr/lib/cups/backend/snmp) |& tee snmp.log
```

On MacOS X you'll find the SNMP backend in /usr/libexec/cups/backend instead:

```
CUPS_DEBUG_LEVEL=2 /usr/libexec/cups/backend/snmp 2>&1 | tee snmp.log
```

The output will look something like this:

```
 1  INFO: Using default SNMP Address @LOCAL
```

```
INFO: Using default SNMP Community public
DEBUG: Scanning for devices in "public" via "@LOCAL"...
DEBUG: 0.000 Sending 46 bytes to 192.168.2.255...
DEBUG: SEQUENCE 44 bytes
DEBUG:     INTEGER 1 bytes 0
DEBUG:     OCTET STRING 6 bytes "public"
DEBUG:     Get-Request-PDU 31 bytes
DEBUG:         INTEGER 4 bytes 1149539174
DEBUG:         INTEGER 1 bytes 0
DEBUG:         INTEGER 1 bytes 0
DEBUG:         SEQUENCE 17 bytes
DEBUG:             SEQUENCE 15 bytes
DEBUG:                 OID 11 bytes .1.3.6.1.2.1.25.3.2.1.2.1
DEBUG:                 NULL VALUE 0 bytes
DEBUG: 0.001 Received 55 bytes from 192.168.2.229...
DEBUG: community="public"
DEBUG: request-id=1149539174
DEBUG: error-status=0
DEBUG: SEQUENCE 53 bytes
DEBUG:     INTEGER 1 bytes 0
DEBUG:     OCTET STRING 6 bytes "public"
DEBUG:     Get-Response-PDU 40 bytes
DEBUG:         INTEGER 4 bytes 1149539174
DEBUG:         INTEGER 1 bytes 0
DEBUG:         INTEGER 1 bytes 0
DEBUG:         SEQUENCE 26 bytes
DEBUG:             SEQUENCE 24 bytes
DEBUG:                 OID 11 bytes .1.3.6.1.2.1.25.3.2.1.2.1
DEBUG:                 OID 9 bytes .1.3.6.1.2.1.25.3.1.5
DEBUG: add_cache(addr=0xbfffe170, addrname="192.168.2.229",
uri="(null)", id="(null)", make_and_model="(null)")
DEBUG: 0.002 Sending 46 bytes to 192.168.2.229...
DEBUG: SEQUENCE 44 bytes
DEBUG:     INTEGER 1 bytes 0
DEBUG:     OCTET STRING 6 bytes "public"
DEBUG:     Get-Request-PDU 31 bytes
DEBUG:         INTEGER 4 bytes 1149539175
DEBUG:         INTEGER 1 bytes 0
DEBUG:         INTEGER 1 bytes 0
DEBUG:         SEQUENCE 17 bytes
DEBUG:             SEQUENCE 15 bytes
DEBUG:                 OID 11 bytes .1.3.6.1.2.1.25.3.2.1.3.1
DEBUG:                 NULL VALUE 0 bytes
DEBUG: 0.003 Received 69 bytes from 192.168.2.229...
DEBUG: community="public"
DEBUG: request-id=1149539175
DEBUG: error-status=0
DEBUG: SEQUENCE 67 bytes
DEBUG:     INTEGER 1 bytes 0
DEBUG:     OCTET STRING 6 bytes "public"
DEBUG:     Get-Response-PDU 54 bytes
DEBUG:         INTEGER 4 bytes 1149539175
DEBUG:         INTEGER 1 bytes 0
DEBUG:         INTEGER 1 bytes 0
DEBUG:         SEQUENCE 40 bytes
DEBUG:             SEQUENCE 38 bytes
DEBUG:                 OID 11 bytes .1.3.6.1.2.1.25.3.2.1.3.1
DEBUG:                 OCTET STRING 23 bytes "HP LaserJet 4000
Series"
DEBUG: 1.001 Probing 192.168.2.229...
DEBUG: 1.001 Trying socket://192.168.2.229:9100...
DEBUG: 192.168.2.229 supports AppSocket!
DEBUG: 1.002 Scan complete!
network socket://192.168.2.229 "HP LaserJet 4000 Series"
"HP LaserJet 4000 Series 192.168.2.229" ""
```

Dissecting the Output

The first two lines are just informational and let you know that the default community name and address are being used. Lines 3-15 contain the initial SNMP query for the device type OID (.1.3.6.1.2.1.25.3.2.1.2.1) from the Host MIB.

Lines 16-31 show the response we got from an HP LaserJet 4000 network printer. At this point we discover that it is a printer device and then send another SNMP query (lines 32-43) for the device description OID (.1.3.6.1.2.1.25.3.2.1.3.1) from the Host MIB as well.

Lines 44-58 show the response to the device description query, which tells us that this is an HP LaserJet 4000 Series printer.

On line 59 we start our active connection probe and discover that this print server supports the AppSocket (JetDirect) protocol on port 9100.

Finally, line 63 shows the device information line for the print server that is sent to CUPS.

Reporting Problems

If you don't see your printer listed, or the wrong information is listed, then you need to gather more information on the printer. The easiest way to do this is to run the snmpwalk command:

```
snmpwalk -Cc -v 1 -c public ip-address | tee snmpwalk.log
```

where "ip-address" is the IP address of the printer or print server. You should see a *lot* of values stream by - the ones you want to see are:

```
HOST-RESOURCES-MIB::hrDeviceType.1 = OID: HOST-RESOURCES-TYPES::hrDevicePrinter
HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: HP LaserJet 4000 Series
```

The hrDeviceType line should show hrDevicePrinter; if not, then your printer or print server doesn't identify itself as a printer. The hrDeviceDescr line should provide a human-readable string for the make and model of the printer, although in some cases you'll just see something less useful like "Axis OfficeBASIC Parallel Print Server".

Once you have collected the snmpwalk output, you should go to the CUPS Bugs & Features page to submit a feature request to support your printer or print server. Be sure to attach those two log files you created - they will help us to identify the SNMP values we need to look for.

Configuring Print Servers

Configuring Axis Print Servers

The Axis print servers can be configured using BOOTP or DHCP. However, on models that do not provide IPP support an additional step must be performed to configure the TCP/IP portion of the print server for use with CUPS.

Each print server contains a configuration file named *config* that contains a list of network parameters used by the server. To modify this file you must first download it from the print server using the `ftp(1)` program:

```
ftp ip-address
Connected to ip-address.
220 Axis NPS ### FTP Printer Server V#.## MON DD YEAR ready.
```

```
ftp> user root
331 User name ok, need password
Password: pass (this is not echoed)
230 User logged in
ftp> get config
local: config remote: config
200 PORT command successful.
150 Opening data connection for config (192,0,2,2),
(mode ascii).
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> quit
221 Goodbye.
```

Next, edit the file with your favorite text editor and locate the lines beginning with:

```
RTN_OPT.     : YES
RTEL_PR1.    : 0
RTEL_PR2.    : 0
RTEL_PR3.    : 0
RTEL_PR4.    : 0
RTEL_PR5.    : 0
RTEL_PR6.    : 0
RTEL_PR7.    : 0
RTEL_PR8.    : 0
```

Change the `RTN_OPT` line to read:

```
RTN_OPT.     : NO
```

This disables the Reverse TELNET protocol and enables the standard TELNET protocol on the print server. Next, assign a port number for each parallel and serial port on the server as follows:

```
RTEL_PR1.    : 9100
RTEL_PR2.    : 9101
RTEL_PR3.    : 9102
RTEL_PR4.    : 9103
RTEL_PR5.    : 9104
RTEL_PR6.    : 9105
RTEL_PR7.    : 9106
RTEL_PR8.    : 9107
```

This essentially makes the Axis print server look like a Hewlett Packard JetDirect EX print server. Save the file and then upload the new *config* file using the `ftp` command:

```
ftp ip-address
Connected to ip-address.
220 Axis NPS ### FTP Printer Server V#.## MON DD YEAR ready.
ftp> user root
331 User name ok, need password
Password: pass (this is not echoed)
230 User logged in
ftp> put config CONFIG
local: config remote: CONFIG
200 PORT command successful.
150 Opening data connection for config (192,0,2,2), (mode ascii).
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> get hardreset
local: hardreset remote: hardreset
200 PORT command successful.
421 Axis NPS ### hard reset, closing connection.
ftp> quit
```

```
221 Goodbye.
```

Your Axis print server is now ready for use!

Configuring Linksys Print Servers

The Linksys print servers can be configured using BOOTP or DHCP. Like older Axis print servers, an additional step must be performed to configure the TCP/IP portion of the print server for use with CUPS.

Each print server contains a configuration file named *CONFIG* that contains a list of network parameters used by the server. To modify this file you must first download it from the print server using the `ftp(1)` program:

```
ftp -n ip-address
Connected to ip-address.
220 Print Server Ready.
Remote system type is Print.
ftp> get CONFIG
local: CONFIG remote: CONFIG
200 Command OK.
150 Open ASCII Mode Connection.
WARNING! 68 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> quit
221 Goodbye.
```

Next, edit the file with your favorite text editor and locate the lines beginning with:

```
0100 L1_PROUT:P1
0120 L2_PROUT:P1
0140 L3_PROUT:P1
```

Change the port number for each parallel and serial port on the server as follows:

```
0100 L1_PROUT:P1
0120 L2_PROUT:P2
0140 L3_PROUT:P3
```

This maps each virtual printer with a physical port. Save the file and then upload the new *CONFIG* file using the `ftp` command:

```
ftp -n ip-address
Connected to ip-address.
220 Print Server Ready.
Remote system type is Print.
ftp> put CONFIG
local: CONFIG remote: CONFIG
200 Command OK.
150 Open ASCII Mode Connection.
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> quit
221 Goodbye.
```

Your Linksys print server is now ready for use!